

## **Rancang Bangun Program Aplikasi Pencetak Kartu Mahasiswa**

**Jati Sasongko**

Fakultas Teknologi Informasi, Universitas Stikubank Semarang

email : jati@unisbank.ac.id

**ABSTRAK :** Mahasiswa sebuah perguruan tinggi diberikan identitas sendiri untuk membedakan dengan mahasiswa perguruan tinggi yang lain. Untuk membedakan mahasiswa dari sebuah perguruan tinggi berdasarkan dari kartu mahasiswanya. Banyaknya mahasiswa yang aktif akan mempengaruhi jumlah kartu identitas yang dibagikan kepada mahasiswa. Mahasiswa akan menggunakan kartu mahasiswa tersebut selama masa studi belum selesai. Apabila kartu mahasiswa dalam masa studi tersebut mengalami kerusakan atau kehilangan maka mahasiswa tersebut wajib mengurus untuk mendapatkan kembali kartu mahasiswanya. Hanya aja apabila terjadi kerusakan atau kehilangan yang jumlahnya sangat sedikit dan waktu yang tidak bersamaan akan mengakibatkan proses pembuatan kartu mahasiswa tidak lagi efektif dan efisien. Dengan kondisi tersebut sangat tepat apabila sebuah perguruan tinggi mempunyai system pencetak kartu sendiri. Adanya system tersebut proses pembuatan kartu yang sangat random dalam masalah jumlah dan waktu dapat teratasi. Model yang ideal untuk membuat aplikasi pencetak kartu mahasiswa menggunakan aplikasi berbasis pada database sistem. Dengan model tersebut apabila sering terjadi proses pengurusan kartu dapat dengan mudah dan cepat dilakukan. Perumusan masalah yang didapatkan bagaimana merancang program untuk mencetak kartu mahasiswa dengan menggunakan barcode sebagai identifikasi kartu mahasiswa? karena masing-masing mahasiswa akan mendapatkan satu kartu mahasiswa. Program aplikasi yang di bangun ini menggunakan bahasa pemrograman Delphi dan SQL Server sebagai sistem basis datanya. Program aplikasi ini dapat digunakan menambah, mengedit, menghapus dan mencari data mahasiswa. Kartu mahasiswa di cetak menggunakan Quick Report. Tujuan yang dapat dicapai adalah menghasilkan suatu program aplikasi yang dapat untuk mencetak kartu mahasiswa dengan ada gambar foto mahasiswa dan barcode sebagai identifikasi kartu.

**Kata kunci :** id card, kartu identitas, mencetak

### **PENDAHULUAN**

Setiap perguruan tinggi mempunyai identitas sendiri untuk membedakan dengan perguruan tinggi yang lain. Mahasiswa sebuah perguruan tinggi juga diberikan identitas sendiri untuk membedakan dengan mahasiswa perguruan tinggi yang lain. Untuk membedakan mahasiswa dari sebuah perguruan tinggi berdasarkan dari kartu mahasiswanya. Jumlah mahasiswa untuk sebuah perguruan tinggi sangat relatif tergantung dari penerimaan mahasiswa baru tiap tahunnya. Ada perguruan tinggi yang mempunyai mahasiswa aktifnya berjumlah ratusan saja tapi ada juga perguruan tinggi yang mahasiswanya sampai ribuan.

Banyaknya mahasiswa yang aktif akan mempengaruhi jumlah kartu identitas yang dibagikan kepada mahasiswa. Mahasiswa akan menggunakan kartu mahasiswa tersebut selama masa studi belum selesai. Apabila kartu

mahasiswa dalam masa studi tersebut mengalami kerusakan atau kehilangan maka mahasiswa tersebut wajib mengurus untuk mendapatkan kembali kartu mahasiswanya.

Kartu mahasiswa dalam jumlah sedikit atau banyak dapat dipesankan di luar perguruan tinggi yang mempunyai wewenang untuk membuatnya. Hanya aja apabila terjadi kerusakan atau kehilangan yang jumlahnya sangat sedikit dan waktu yang tidak bersamaan akan mengakibatkan proses pembuatan kartu mahasiswa tidak lagi efektif dan efisien.

Dengan kondisi tersebut sangat tepat apabila sebuah perguruan tinggi mempunyai system pencetak kartu sendiri. Adanya system tersebut proses pembuatan kartu yang sangat random dalam masalah jumlah dan waktu dapat teratasi.

Model yang ideal untuk membuat aplikasi pencetak kartu mahasiswa menggunakan aplikasi berbasis pada database sistem. Dengan

model tersebut apabila sering terjadi proses pengurusan kartu dapat dengan mudah dan cepat dilakukan.

## **PERUMUSAN MASALAH**

Pencetakan kartu mahasiswa sangat berhubungan dengan data mahasiswa yang masih aktif, sehingga didapat permasalahan bagaimana merancang program untuk mencetak kartu mahasiswa dengan menggunakan barcode sebagai identifikasi kartu mahasiswa? karena masing-masing mahasiswa akan mendapatkan satu kartu mahasiswa.

## **BATASAN MASALAH**

Program aplikasi yang di bangun ini menggunakan bahasa pemrograman Delphi dan SQL Server sebagai sistem basis datanya. Program aplikasi ini dapat untuk menambah, mengedit, menghapus dan mencari data mahasiswa. Kartu mahasiswa di cetak menggunakan Quick Report.

## **TUJUAN**

Tujuan penelitian yang diharapkan bisa tercapai adalah menghasilkan suatu program aplikasi yang dapat untuk mencetak kartu mahasiswa dengan ada gambar foto mahasiswa dan barcode sebagai identifikasi kartu.

## **METODE PENGEMBANGAN SISTEM**

Metode pengembangan sistem yang di gunakan ada beberapa tahap antara lain

### **1. Tahap Perencanaan Sistem.**

Tahap dalam mengenali, mendiagnosa dan mendefinisikan masalah kegiatan yang dilakukan meliputi :

- a. Merumuskan masalah dengan mendefinisikan batasan masalah dan pokok permasalahan serta menentukan tujuan dan manfaat penelitian.
- b. Menentukan prioritas penanganan masalah pada analisa OOD (Object Oriented Development) dan perancangan program aplikasi pencetakan kartu mahasiswa.

### **2. Tahap Analisa Sistem.**

Tahap untuk menganalisa dan mengamati sistem secara lebih rinci digunakan sebagai dasar dalam menyusun dan merancang sistem baru. Kegiatan yang dilakukan :

- a. Mempelajari system pencetakan kartu mahasiswa.
- b. Mempelajari permasalahan pada sistem lama untuk diperbaiki dan dikembangkan.
- c. Mengembangkan alternatif pemecahan masalah.

### **3. Tahap Desain Sistem.**

Tahap desain sistem merupakan tahap penyusunan sistem baru dan menuangkannya secara tertulis. Kegiatan yang dilakukan :

- a. Mendesain format sistem baru berdasarkan sistem lama yang telah diperbaiki.
- b. Merencanakan konfigurasi pemakai yang diperlukan dalam pelaksanaan sistem.

### **4. Tahap Rencana Implementasi Sistem.**

Tahap rencana implementasi merupakan tahap pengalihan sistem lama ke sistem baru. Kegiatan yang dilakukan meliputi:

- a. Menentukan orang yang akan mengoperasikan komputer.
- b. Menentukan perangkat keras (hardware) yang dibutuhkan.
- c. Menentukan perangkat lunak (software) yang dibutuhkan.

## **ANALISA DAN PERANCANGAN**

Aplikasi pencetak kartu mahasiswa di rancang dengan menggunakan metode object oriented development. Diagram-diagram yang digunakan untuk merancang meliputi use case diagram, class diagram, sequence diagram dan state diagram. Diagram yang telah dibuat akan diterjemahkan ke dalam bahasa pemrograman delphi. Selain bahasa pemrograman dephi juga ada sql server sebagai database servernya.

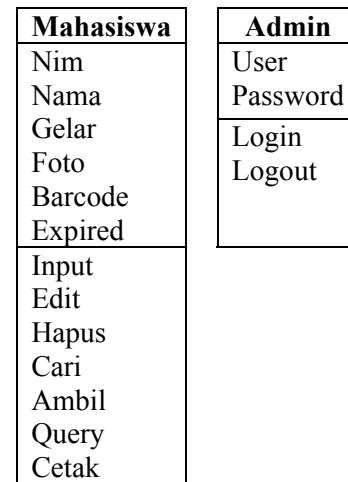
### **Use Case Diagram**

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Dari gambar 1 dapat dijelaskan bahwa aktor dari program aplikasi ini merupakan seorang admin, yang mempunyai tugas untuk mengolah data. Di mana mengolah data meliputi input data, edit data, hapus data hingga mencetak data. Ada juga fasilitas untuk mencari data dengan model query dimana fungsinya untuk mencari data berdasarkan kriteria tertentu. Untuk dapat masuk pada aplikasi ini diperlukan validasi berupa memasukkan user dan password, sehingga keamanan datanya akan lebih aman. Sedangkan untuk mencetak data, dapat dilakukan secara kolektif atau personal tergantung dari data yang akan dicetak.

### Class Diagram

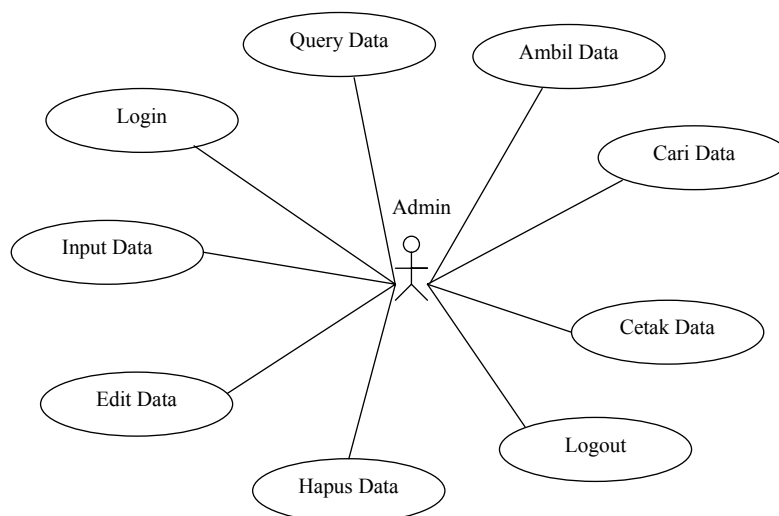
Class diagram merupakan diagram paling umum dipakai di semua pemodelan berorientasi objek. Pemodelan kelas merupakan pemodelan paling utama di pendekatan berorientasi objek pemodelan kelas menunjukkan kelas – kelas



Gambar 2. Class Diagram

yang ada di system dan hubungan antar kelas – kelas itu, atribut – atribut dan operasi – operasi di kelas – kelas. Kelas di diagram kelas dapat secara langsung diimplementasikan di bahasa pemrograman berorientasi objek yang secara langsung mendukung bentukan kelas.

Dari gambar 2 menggambarkan bahwa pada aplikasi ini terdapat dua class yaitu mahasiswa dan admin. Dari class mahasiswa didapatkan variabel yang akan diolah berupa nim, nama, gelar, foto, barcode dan expired. Variabel-variabel tersebut dapat diolah dengan cara di input, edit, hapus, cari, ambil, query dan cetak. Variabel yang lain didapatkan dari class admin yaitu user dan password, dimana user dan password diproses dengan menggunakan login dan logout.



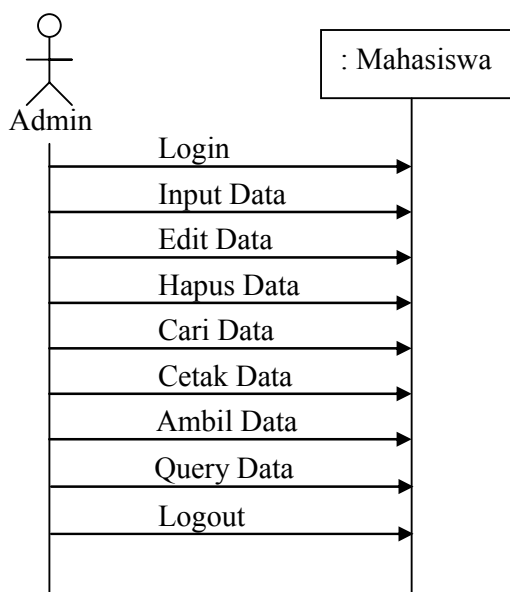
Gambar 1. Use Case Diagram

## Sequence Diagram

Sequence diagram digunakan untuk memodelkan skenario penggunaan. Skenario penggunaan adalah barisan kejadian yang terjadi selama satu eksekusi system. Cakupan skenario dapat beragam, dari mulai semua kejadian di system atau hanya kejadian pada objek – objek tertentu. Skenario menjadi rekaman historis eksekusi system atau gagasan eksperimen eksekusi system yang diusulkan.

Sequence diagram menunjukkan objek sebagai garis vertical dan tiap kejadian sebagai panah horizontal dari objek pengirim ke objek penerima. Waktu berlaku dari atas ke bawah dengan lama waktu tidak relevan. Diagram ini hanya menunjukkan barisan kejadian, bukan pewaktuan nyata. Kecuali untuk system waktu nyata yang mengharuskan konstrain barisan kejadian. Masing-masing objek, termasuk aktor, memiliki lifeline vertikal.

Dari gambar 3 dapat digambarkan bahwa aktor dari sistem ini adalah seorang admin, sedangkan objek dari sistem ini adalah mahasiswa, hanya saja mahasiswa yang dimaksudkan bukan sebagai individu tetapi sebagai objek data. Seorang admin dapat melakukan input, edit, hapus dan sebagainya terhadap objek mahasiswa, tidak berdasarkan urutan proses, tetapi lebih ditekankan kepada apa saja yang dapat dilakukan oleh admin terhadap objek mahasiswa dalam satuan waktu



Gambar 3. Sequence Diagram

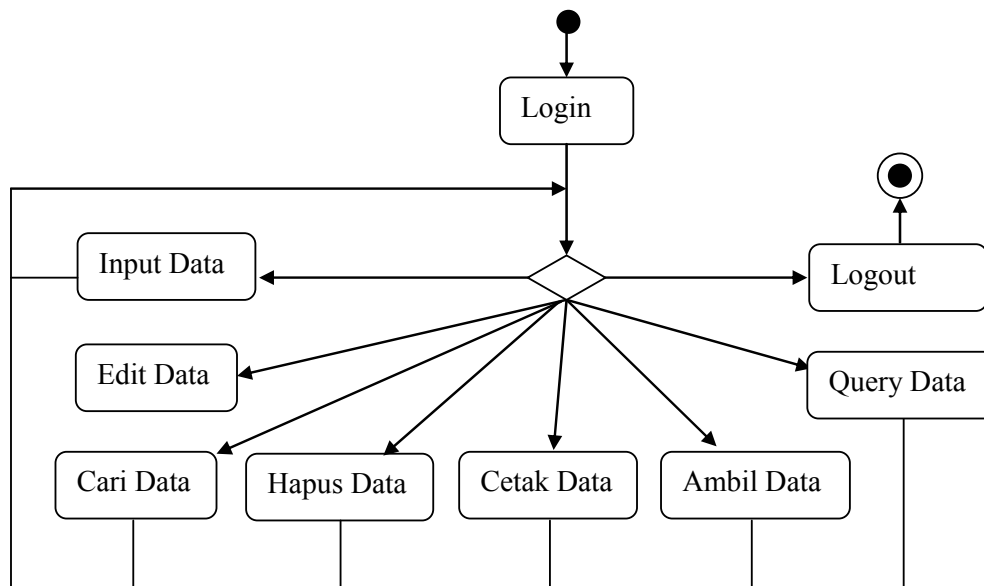
tertentu. Hanya saja proses login dan logout pasti dilakukan terlebih dahulu dan terakhir apabila akan menjalankan sistem ini.

## Statechart Diagram

Statechart diagram memperlihatkan berbagai state (keadaan sesaat) yang dilalui sebuah obyek, dan kejadian – kejadian yang menyebabkan sebuah transisi dari satu state ke state yang lainnya, dan aksi yang mengakibatkan suatu perubahan state. (Suhendar dan Gunadi, 2002). Pola kejadian, state dan transisi untuk satu kelas dapat diabstraksikan dan direpresentasikan dengan diagram state (state diagram). Statechart merupakan perluasan diagram state. Statechart mempermudah pemodelan perilaku sistem secara tidak ambigu. Statechart menyediakan notasi dan sekumpulan konvensi fasilitas dekomposisi hirarki mesin-mesin finite state dan mekanisme berkomunikasi di antara mesin-mesin finite state konkuren.

Satu diagram state (statechart) untuk satu kelas objek dimana perilaku dinamisnya penting dan menunjukkan pola aktivitas. Masing-masing mesin state berjalan secara konkuren dan dapat mengubah state secara independen. Diagram-diagram state beragam kelas membentuk satu model dinamis tunggal lewat kejadian bersama (*shared events*).

Dari gambar 4 dapat dijelaskan bahwa aplikasi ini selalu dimulai dengan memasukkan login sebagai sarana validasi untuk keamanan data. Ketika login tersebut benar maka akan ditampilkan program secara keseluruhan, dimana terdapat beberapa pilihan proses yang dapat dilakukan. Pilihan proses tidak menunjukkan urutan proses tetapi tergantung dari kebutuhan yang akan dilakukan. Setiap akhir menjalankan aplikasi selalu dilakukan proses logout, yaitu keluar dari aplikasi sekaligus database-nya.



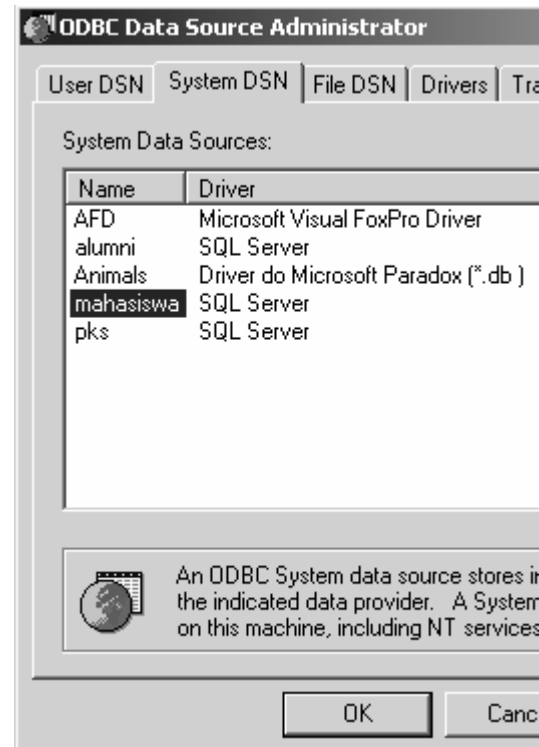
Gambar 4. Statechart diagram

## Rancangan Table

[illegible]

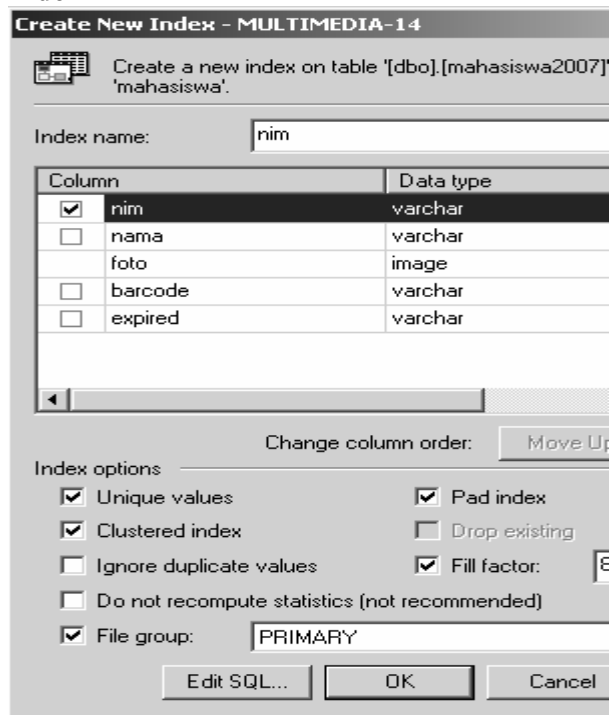
Gambar 5. Rancangan table mahasiswa

## Koneksi Database



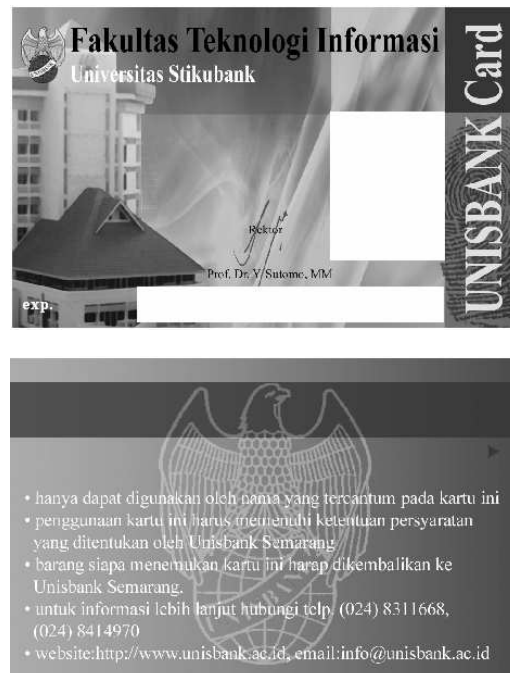
Gambar 6. Koneksi database ODBC

## Index



Gambar 7. Membuat index

## Rancangan Tampilan Kartu



Gambar 8. Rancangan tampilan kartu



Gambar 9. Rancangan tampilan program

**Contoh prosedur-prosedur dalam aplikasi :**

```

procedure TForm1.Cari (Sender: TObject);
begin
  Query1.DatabaseName := 'mahasiswa';
  Query1.Close;
  Query1.SQL.Clear;
  Query1.SQL.Add('select      *      from      '+'
  ComboBox4.Text  +' where  nim  like
  '%" + Edit1.Text + '%"');
  Query1.Open;
end;

```

```

Query
procedure TForm1.Query (Sender: TObject);
begin
  try
  begin
    Query1.DatabaseName := 'mahasiswa';
    Query1.SQL.Clear;
    Query1.SQL.Add(Edit2.Text);
    Query1.Open;
  end;
  except
  begin
    beep;
    showmessage('Pernyataan SQL tidak sah');
  end;
  end;
end;

```

```

procedure TForm1.Ambil (Sender: TObject);
var
  ada : boolean;
  n : string;
begin
  n := DBEdit1.Text;
  ada := table2.findkey([n]);
  if ada then
    showmessage ('Data sudah ada')
  else
    begin
      Table2.Open;
      Table2.Append;
      Table2['nim']:=DBEdit1.Text;
      Table2['nama']:=DBEdit2.Text;
      Table2['barcode']:=DBEdit3.Text;
      Table2['expired']:=DBEdit4.Text;
      DBImage2.Picture:=DBImage1.Picture;
      Table2.Post;
      Table2.Close;
      Table2.Open;
    end;
  end;

```

```

end;
end;

```

```

procedure TForm1.Semua (Sender: TObject);
begin
  DataSource1.DataSet.First;
  repeat
    Table2.Append;
    Table2['nim']:=DBEdit1.Text;
    Table2['nama']:=DBEdit2.Text;
    Table2['barcode']:=DBEdit3.Text;
    Table2['expired']:=DBEdit4.Text;
    DBImage2.Picture:=DBImage1.Picture;
    Table2.Post;
  until DataSource1.DataSet.Next;
  until DataSource1.DataSet.Eof;
  Table2.Close;
  Table2.Open;
end;

```

```

procedure  TForm1.HapusSemua    (Sender:
TObject);
begin
  Table2.First;
  repeat
    Table2.Delete;
  until Table2.Eof;
end;

```

```

procedure TForm1.Load (Sender: TObject);
begin
  DataSource1.Edit;
  if OpenPictureDialog1.Execute Then
    DBImage1.Picture.LoadFromFile(OpenPictureD
ialog1.FileName);
end;

```

```

procedure TForm1.Cetak (Sender: TObject);
begin
  if ComboBox5.Text = 'FEA' then
    begin
      Form2.QuickRep1.Preview;
      Form2.Table1.Close;
      Form2.Table1.Open;
      Form2.Table1.Refresh;
    end
  else
    if ComboBox5.Text = 'FTT' then
      begin
        Form3.QuickRep1.Preview;
        Form3.Table1.Close;
        Form3.Table1.Open;
      end;
    end;

```

```

Form3.Table1.Refresh;
end
else
if ComboBox5.Text = 'FEB' then
begin
Form4.QuickRep1.Preview;
Form4.Table1.Close;
Form4.Table1.Open;
Form4.Table1.Refresh;
end
else
if ComboBox5.Text = 'FH' then
begin
Form5.QuickRep1.Preview;
Form5.Table1.Close;
Form5.Table1.Open;
Form5.Table1.Refresh;
end
else
if ComboBox5.Text = 'FT' then
begin
Form6.QuickRep1.Preview;
Form6.Table1.Close;
Form6.Table1.Open;
Form6.Table1.Refresh;
end

```

```

else
if ComboBox5.Text = 'FBIB' then
begin
Form7.QuickRep1.Preview;
Form7.Table1.Close;
Form7.Table1.Open;
Form7.Table1.Refresh;
end
else
if ComboBox5.Text = 'FA' then
begin
Form8.QuickRep1.Preview;
Form8.Table1.Close;
Form8.Table1.Open;
Form8.Table1.Refresh;
end
end;

```

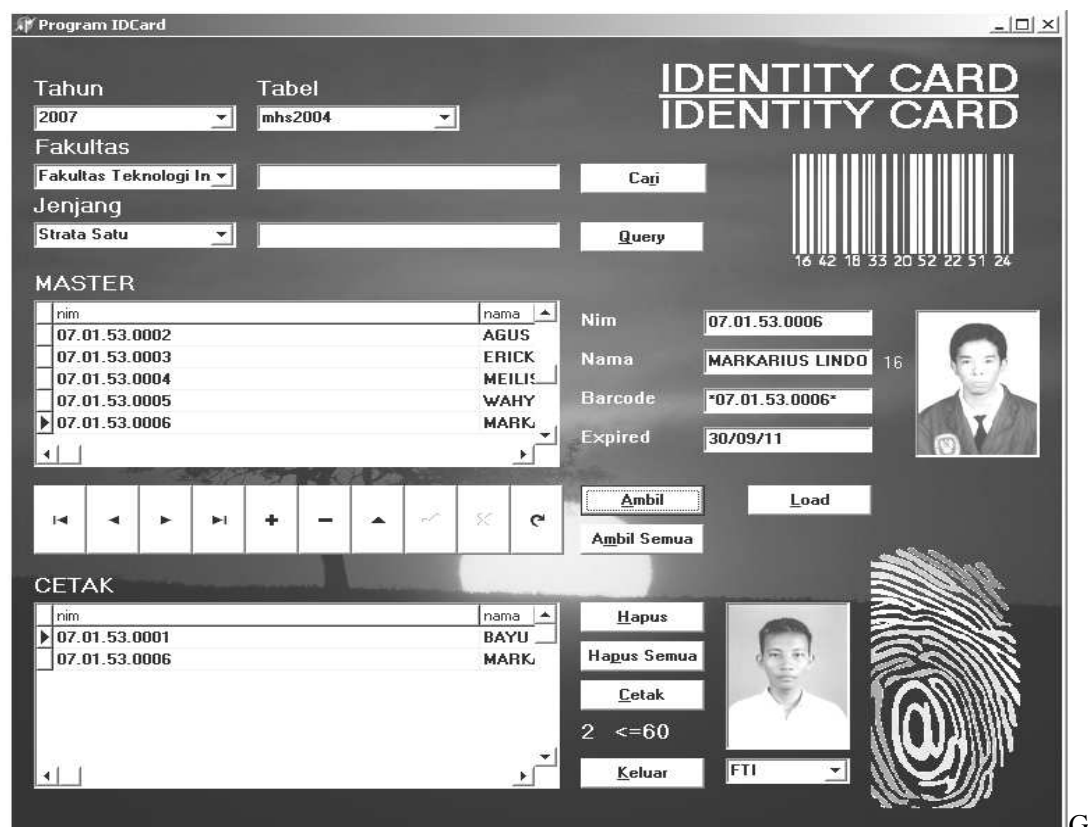
```

procedure TForm1.Keluar (Sender: TObject);
begin
Application.Terminate;
end;

```

## IMPLEMENTASI

### Tampilan Program



Gambar 10. Tampilan program



### Tampilan Depan Kartu Mahasiswa



Gambar 11. Tampilan depan kartu mahasiswa

### Tampilan Belakang Kartu Mahasiswa



Gambar 12. Tampilan belakang kartu

### KESIMPULAN

Adapun dari hasil yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut:

1. Kelemahan dari kondisi sistem lama yaitu tidak adanya aplikasi khusus untuk pencetak kartu mahasiswa, yang ada hanya pengolah data mahasiswa, sehingga dengan adanya aplikasi ini dapat digunakan untuk mencetak kartu mahasiswa.
2. Dalam mencetak kartu mahasiswa masih terbatas dalam 60 record data, sehingga perlu dikembangkan lagi untuk dapat mencetak berapun jumlah yang diinginkan.
3. Untuk memasukkan data foto mahasiswa dibutuhkan alat scanner, akan lebih mudah apabila nantinya dikembangkan lagi dengan menggunakan kamera digital.

### DAFTAR PUSTAKA

1. Booch G., 1994, *Object - Oriented Analysis And Design*, The Benjamin / Cummings Publishing Company, Inc, California
2. Dennis De Champeaux, Dauglas Lea and Penelope Faure, 1994, *Object Oriented System Development*, Addison Wesley Publishing Company, California
3. Jogyanto H. M, 1999, *Analisa dan Desain Sistem Infomasi*
4. Kristanto H., Ir, 1993, *Konsep dan Perancangan Data Base*, Andi Offset, Yogyakarta.
5. Matcho J., Faulker D.R., 1997, *Panduan Penggunaan Delphi*, Andi, Yogyakarta
6. Nugroho A., 2002, *Metodologi Berorientasi Objek*, 2002
7. Pramono D., 1996, *Contoh Penggunaan Rutin - Rutin Borland Delphi*, PT. Elex Media Komputindo, Jakarta
8. Pranata A., 1998, *Pemrograman Borland Delphi*, Andi, Yogyakarta
9. Suhendar A., Guradi H., 2002, *Visual Modeling Menggunakan UML dan Rational Rose*
10. Sutanta E., 1996, *Sistem Basis Data*, Andi Offset, Yogyakarta